

## Техническое задание (БА+СА) аналитик

### Задача 3

#### Задание 3.1: REST API: Регистрация пользователя

1. http метод + URL: POST /api/register

2. Входные параметры (Request):

Название	Тип	Обязательность	Ограничения
firstName	string	да	1-30 символов, только буквы
lastName	string	да	1-30 символов, только буквы
username	string	да	3-30 символов, только буквы, должен быть уникальным
password	string	да	Минимум 8 символов, минимум 1 цифра, 1 заглавная буква, 1 строчная буква, 1 спецсимвол
captchaToken	string	да	Непустая строка, валидный токен captcha

Пример запроса:

```
{  
  "firstName": "Ivan",  
  "lastName": "Ivanov",  
  "username": "ivan",  
  "password": "Qwerty123!",  
  "captchaToken": "captcha-value"  
}
```



3. Выходные параметры при успешном ответе:

HTTP 201 Created

Название	Тип	Обязательность	Ограничения
id	string	да	Уникальный идентификатор пользователя (UUID), не пустой
username	string	да	3–30 символов, латиница, соответствует логину пользователя
message	string	да	Непустая строка, фиксированный текст: User registered successfully

Пример ответа для успешного сценария:

```
{  
  "id": "12345",  
  "username": "ivan",  
  "message": "User registered successfully"  
}
```

4. Выходные параметры при ответе с ошибкой:

Название	Тип	Обязательность	Ограничения
message	string	да	Непустая строка, читаемое описание ошибки
errorCode	string	нет	Код ошибки из фиксированного набора (например: VALIDATION_ERROR, USER_ALREADY_EXISTS,



			CAPTCHA_REQUIRED, INTERNAL_ERROR)
--	--	--	--------------------------------------

#### 5. Описание ошибок - коды ответов:

- 400 Bad Request — ошибка клиента. Используется если данные введены неправильно (например, слабый пароль)

Текст сообщения:

```
{
  "message": "Password does not meet requirements"
}
```

- 409 Conflict — ошибка клиента. Используется если пользователь уже существует.

Текст сообщения:

```
{
  "message": "User already exists"
}
```

- 403 Forbidden — ошибка клиента. Используется если не пройдена captcha.

Текст сообщения:

```
{
  "message": "Please verify captcha"
}
```

- 500 Internal Server Error — ошибка сервера. Используется при внутренней ошибке сервера.

Текст сообщения:

```
{
  "message": "Internal server error"
}
```

Фронтенд отправляет POST-запрос на /api/register с данными пользователя. Бэкенд валидирует данные, проверяет captcha и уникальность username. Если всё хорошо — возвращает 201. Если ошибка пользователя — 400, если проблема сервера — 500.



Задание 3.2 Сначала бэк проверяет входные данные, потом captcha, затем уникальность пользователя. Если все проверки прошли — пользователь сохраняется в базе и возвращается успешный ответ. При любой ошибке возвращается соответствующий HTTP-код.

1. После нажатия кнопки Register фронтенд отправляет запрос: POST /api/register с данными пользователя.
2. Бэк-сервис принимает запрос и проверяет, что он пришёл в формате JSON (Content-Type: application/json).

Если формат неправильный — возвращает ошибку 400 Bad Request.

3. Бэк-сервис проверяет, что в запросе переданы все обязательные поля: firstName, lastName, username, password, captchaToken.

Если какого-то поля нет или оно пустое — возвращает 400 Bad Request.

4. Бэк-сервис валидирует данные: имя и фамилия — строка нужной длины; username — корректный формат; пароль — не менее 8 символов и соответствует требованиям

Если данные не валидны — возвращает 400 Bad Request.

5. Бэк-сервис проверяет captcha, отправляя captchaToken во внешний сервис.

Если captcha не пройдена — возвращает 403 Forbidden.

6. Бэк-сервис проверяет в базе данных, существует ли пользователь с таким username.

Если пользователь уже есть — возвращает 409 Conflict.

7. Если все проверки прошли успешно, бэк-сервис: генерирует уникальный id пользователя; хэширует пароль; подготавливает данные пользователя для сохранения.
8. Бэк-сервис сохраняет пользователя в базе данных.

Если при сохранении произошла ошибка — возвращает 500 Internal Server Error.

9. Если пользователь успешно сохранён, бэк-сервис возвращает ответ: 201 Created с id пользователя и сообщением об успешной регистрации.